Goodbye fat gem 2025

Sutou Kouhei

ClearCode Inc.

RubyKaigi 2025

2025-04-16

Sutou Kouhei
A president Ruby committer

The president of ClearCode Inc.

クリアコードの社長

Silver Sponsors

ClearCode Inc.

https://www.clear-code.com/

Free software is important in ClearCode. We develop/support software with our free software development experiences. We feed back our business experiences to free software.

The update of "Goodbye fat gem"

Goodbye fat gem

Sutou Kouhei

ClearCode Inc.

RubyKaigi Takeout 2020
2020-09-04

Fat gem a.k.a. {binary,native} gem

ClearCode

Gem that includes pre-built binaries

ビルド済みバイナリー入りのgem

Why is fat gem needed?
どうしてfat gemが必要なのか

- ✓ Difficult to build extension library
  拡張ライブラリーのビルドが難しい
  - ✓ Extension library:
    A Ruby library implemented with C API
    拡張ライブラリー：C APIを使って実装されたRubyライブラリー
- ✓ Fat gem just copies pre-built binaries
  fat gemは単にビルド済みバイナリーをコピーするだけ（ビルドしない）

Why is building it difficult?
どうして拡張ライブラリーのビルドは難しいのか

- Users need build environment
  ユーザーはビルド環境を用意しないといけない
  - e.g.: C/Rust/Go compiler and so on
    例：C/Rust/Goコンパイラーなど
- Users need build dependencies
  ユーザーが依存するライブラリーを用意しないといけない
  - e.g.: GTK 4 for gtk4 gem
    例：gtk4 gemはGTK 4が必要

With fat gem
fat gemを使うと

- ✓ Users don't need build environment!
  ユーザーはビルド環境を用意しなくてもよい
  - ✓ No compiler
    ユーザーはビルド環境がなくてもよい
- ✓ Users don't fail installation!
  ユーザーはインストールに失敗しない
  - ✓ Fat gem just copies pre-built binaries
    fat gemは単にビルド済みバイナリーをコピーするだけ

With fat gem
fat gemを使うと

Happy!

All extension libraries
must be released as fat gems!!!

すべての拡張ライブラリーはfat gemとしてリリースされるべきじゃん！！！

...Really?

…ほんとに？

A fat gem maintainer says...

あるfat gemメンテナー曰く…

Thanks fat gem!
Goodbye fat gem!

ありがとうfat gem！
さよならfat gem！

Fat gem problem 1

fat gemの問題1

Can't use the latest Ruby immediately

いち早く最新のRubyを使えない

Details
詳細

- ✓ Ruby is released every Christmas
  Rubyは毎年クリスマスにリリースされる
- ✓ To use fat gems with the latest Ruby:
  最新のRubyでfat gemを使うには
  - ✓ Need to release fat gems for it
    最新のRuby用のfat gemがリリースされていないといけない
- ✓ Users can't use the latest Ruby while:
  ユーザーは↓の間は最新のRubyを使えないまま
  - ✓ Any of fat gems doesn't support it
    使っているfat gemのどれか1つでも最新のRubyをサポートしていない

The Ruby 3.4 and Nokogiri case

We can use immediately!!!

すぐに使えるじゃん！！！

From extension gem maintainer view

拡張gemメンテナー視点

Not all extension gem maintainers can do it **immediately**

すべての拡張gemメンテナーが**すぐに**対応できるわけじゃない

As a user
ユーザーとして

- ✓ Do you think only people who can release immediately must maintain extension gems?
  すぐにリリースできる人たちだけが拡張gemをメンテナンスするべき？
- ✓ Is it desired Ruby ecosystem?
  そんなRubyエコシステムがよさそう？

Wait!
ちょっと待って！

- Can we fallback to a normal gem until the latest Ruby's fat gem is released?
  最新のRuby用fat gemリリースまでは通常のgemにフォールバックできない？
  - Python's wheel can do it
    Pythonのwheelならできる
  - RubyGems can implement similar approach
    RubyGemsでも同じような仕組みを実装することはできる
- But... do you use it...?
  でも、その仕組み、使う…？

A common scenario
よくあるシナリオ

1. Ruby 3.5 is released!
   Ruby 3.5リリース！
2. Fat gem A isn't 3.5 ready yet...
   fat gemのAはまだRuby 3.5に対応していない…
3. bundle install failed with Ruby 3.5
   Ruby 3.5ではbundle install失敗
   - ✓ Because you don't have build environment...
     ビルド環境を整備していないからなぁ…
4. Pin Ruby to 3.4!
   しばらくRuby 3.4を使っていよう！

Another scenario
別のシナリオ

1. We always use the latest Ruby!
   常に最新のRubyを使うぞ！
2. No fat gem must not block it!
   fat gemの有無で阻害されたくない！
3. Stop using fat gem!
   fat gemを使うのをやめよう！
   - ✓ Prepare build environment
     ビルド環境を整備する
   - ✓ bundle config force_ruby_platform true

FYI: How to implement the fallback

参考情報：フォールバックの実装方法

- ✓ Normal flow:
  通常のフロー
  a. Find nokogiri-X.Y.Z-x86_64-linux.gem (fat gem)
     nokogiri-X.Y.Z-x86_64-linux.gemを探す
  b. Exist: Use it
     あったらそれを使う
  c. Not exist: Use nokogiri-X.Y.Z.gem (normal gem)
     なかったらnokogiri-X.Y.Z.gemを使う
- ✓ Point: The x86_64-linux part
  ポイント：x86_64-linuxのところ

FYI: How to implement the fallback

参考情報：フォールバックの実装方法

- ✓ x86_64-linux
  - ✓ Includes architecture and OS
    アーキテクチャーとOSは含んでいる
  - ✓ But doesn't include Ruby version
    でもRubyのバージョンは含んでいない
- ✓ A fat gem includes binaries for all supported Rubies on the architecture and OS
  fat gemには対象アーキテクチャーのOS用のすべてのサポートしているRuby用のバイナリーが含まれている

FYI: How to implement the fallback
参考情報：フォールバックの実装方法

- RubyGems can't know whether x86_64-linux gem includes binaries for Ruby 3.5 or not
  RubyGemsはx86_64-linux gemがRuby 3.5用のバイナリーを含むかわからない
- How to solve it?
  解決方法は？
  - Add Ruby version: x86_64-linux-ruby3.4, x86_64-linux-ruby3.5, ...
    This is what Python's wheel does
    Rubyのバージョンも入れちゃえばいい（PythonのwheelはこうしてIる）

Fat gem problem 2

fat gemの問題2

Vulnerability response may get delayed

脆弱性対応が遅れがち

Details
詳細

- ✓ Only bindings fat gem case
  バインディングのfat gemだけのケース
  - ✓ Bindings: A extension library to use an external library
    バインディング：外部のライブラリーを使う拡張ライブラリー
  - ✓ Includes an external library binary
    fat gem内に外部ライブラリーのバイナリーも含む
- ✓ When a vulnerability of an external library is found:
  外部ライブラリーに脆弱性が見つかった場合：
  - ✓ Should be released immediately with fix
    すぐに修正を含んだバージョンをリリースするべき

libxslt 1.1.43 + Nokogiri

- ✓ libxslt: 1.1.43: 2025-03-12
  - ✓ CVE-2025-24855, CVE-2024-55549
- ✓ Nokogiri: 1.18.4: **2025-03-14**
  - ✓ Updates bundled libxslt to 1.1.43
- ✓ Debian: bookworm: **2025-03-15**
  - ✓ Fixes CVE-2025-24855, CVE-2024-55549
- ✓ Nokogiri is faster than Debian!!!

Nokogiriの方がDebianより対応が速いじゃん！！！

From fat gem maintainer view

fat gemメンテナー視点

Not all fat gem maintainers can do it **immediately**

すべてのfat gemメンテナーが**すぐに**対応できるわけじゃない

As a user
ユーザーとして

- ✓ Do you think only people who can release immediately must maintain extension gems?
  すぐにリリースできる人たちだけが拡張gemをメンテナンスするべき？
- ✓ Is it desired Ruby ecosystem?
  そんなRubyエコシステムがよさそう？

Fat gem problem 3

fat gemの問題3

High maintenance cost

メンテナンスが大変

Details
詳細

- ✓ Especially binding fat gem case
  特にバインディングのfat gemのケース
- ✓ Normally, cross-compiling is used
  通常、クロスコンパイルしてfat gemを作る
  - ✓ Most external libraries don't do it
    多くの外部ライブラリーはクロスコンパイルなんてしない
  - ✓ There are some problems on upgrading
    外部ライブラリーのバージョンをあげるたびになにかしら問題が見つかる
  - ✓ Ruby-GNOME has 10+ related external libraries
    Ruby-GNOMEは10以上の関連外部ライブラリーがあって大変だった

Wait!
ちょっと待って！

- ✓ Does rake-compiler-dock help us?
  rake-compiler-dockで解決しないの？
  - ✓ We can prepare stable cross-compiling env
    これで安定してクロスコンパイル環境を用意できる
  - ✓ Cross-compiling itself is difficult!!!
    クロスコンパイル自体がキビシイんだよー！！！
- ✓ BTW, do you know all target platforms?
  ところで、すべてのクロスコンパイル対象のプラットフォームを知ってる？

Target platforms
対象プラットフォーム

11 platforms:

- ✓ {x86{,_64},arm,aarch64}-linux-{gnu,musl}
  - ✓ 8: 32/64bit, Intel/ARM, Linux, glibc/musl libc
- ✓ {x86_64,arm64}-darwin
  - ✓ 2: 64bit, Intel/ARM, macOS
- ✓ x64-mingw-ucrt: Windows

Target platforms x Ruby version

対象プラットフォーム × Rubyのバージョン

11 platforms x 4 versions = 44 patterns!

- ✓ {x86{,_64},arm,aarch64}-linux-{gnu,musl}
  - ✓ 8: 32/64bit, Intel/ARM, Linux, glibc/musl libc
- ✓ {x86_64,arm64}-darwin
  - ✓ 2: 64bit, Intel/ARM, macOS
- ✓ x64-mingw-ucrt: Windows

Static linking is difficult too
スタティックリンクも難しいんだよー

- ✓ Binding fat gem uses static linking
  バインディングfat gemはスタティックリンクを使う
  - ✓ May mix multiple different zlib, curl, OpenSSL, ...
    ((複数の違うバージョンのzlib, curl, OpenSSLなどが混在するかも
  - ✓ Symbol conflicts with dlopen(RTLD_GLOBAL)
    dlopen(RTLD_GLOBAL)だとシンボルが衝突するかも
  - ✓ Namespace may fix this?
    Namespaceはdlopen(RTLD_LOCAL)なので解決するかも？

How to solve these problems?

これらの問題を解決するには？

Prepare build env

automatically!?

ビルド環境を**自動で**準備すればよいのでは！？

Predecessor: RubyInstaller2

先駆者：RubyInstaller2

- ✓ Ruby distribution for Windows
  Windows用のRubyディストリビューション
- ✓ Based on MSYS2
  MSYS2ベース
  - ✓ Build environment (Devkit) is included
    Devkitという名前のビルド環境がセット
  - ✓ Package manager is included
    パッケージマネージャもセット

msys2_mingw_dependencies

```
gemspec.metadata["msys2_mingw_dependencies"] = "gtk4"
```

- ✓ gem's metadata for RubyInstaller2
  gemの独自メタデータ
  - ✓ Install dependencies automatically!!!
    依存パッケージを自動インストール！！！
  - ✓ No install failure by missing dependencies!
    依存パッケージがなくてインストール失敗ということはない！

Discussed
相談した

- ✓ "[RFC] Allow specifying and installing external dependencies" in 2022
  https://gist.github.com/postmodern/4c0cbccc0c7eda4585db0fc5267cdd57
  [RFC] 外部依存情報を指定してインストールできるようにしない？

The RFC draft content
このRFCドラフトの内容

- ✓ Problem: Difficult to install gems that have external dependencies
  問題：外部依存があるgemのインストールはむずかしい
- ✓ Proposal:
  - ✓ Allow specifying external deps in gemspec
    gemspecで外部依存を指定できるようにしよう
  - ✓ Install them automatically on install
    インストール時に自動で外部依存もインストールしよう

Who did discuss it?
だれが相談していたの？

- ✓ @postmodern: ruby-install author
- ✓ @flavorjones:
  - ✓ Nokogiri maintainer
  - ✓ rake-compiler-dock maintainer
- ✓ @kou:
  - ✓ native-package-installer author
  - ✓ rake-compiler maintainer

And then?
最終的にどうなったの？

- ✓ No consensus and no action...
  話がまとまらなく具体的なアクションはなかった…
- ✓ But!
  - ✓ Use-cases and requirements were organized
    ユースケースや必要なことは整理された
  - ✓ I got a little motivated💪
    私がちょっとやる気になった

💪 rubygems-requirements-system

A new RubyGems plugin for it:

```
$ gem install rubygems-requirements-system
$ gem install gtk4
```

A new Bundler plugin for it:

```
# Gemfile
plugin "rubygems-requirements-system"
gem "gtk4"
```

How to use by developers
開発者の使い方

```ruby
Gem::Specification.new do |spec|
  # "system" <- Keyword
  # "gtk4" <- Package ID
  # "debian", "homebrew" <- Platform ID
  # "libgtk-4-dev", "gtk4" <- Package name in the platform
  spec.requirements << "system: gtk4: debian: libgtk-4-dev"
  spec.requirements << "system: gtk4: homebrew: gtk4"
  # ...
  # That's all!
end
```

Supported platforms
対応プラットフォーム

Alpine Linux, ALT Linux, Amazon Linux, Arch Linux, Conda, Debian, Fedora, FreeBSD, Gentoo Linux, Homebrew, MacPorts, PLD Linux, RHEL, SUSE and Ubuntu

What problems are solved by this?
これで何が解決するの？

- ✓ Can't use the latest Ruby immediately
  すぐに最新のRubyを使えない
- ✓ Can't update vulnerable products immediately
  すぐに脆弱性のあるプロダクトを更新できない
- ✓ High maintenance cost
  高いメンテナンスコスト

By goodbye fat gem
fat gemをやめることで

What problems are remained?
残る問題はなに？

- ✓ Longer install time
  インストール時間が長くなる
  - ✓ Build time > Copy pre-built binaries time
    ビルド時間 > ビルド済みをコピーする時間
- ✓ Some people don't like system change
  システムが変更されることがイヤな人がいる
  - ✓ e.g.: They don't like apt install XXX
    たとえばapt install XXXがイヤ

How to resolve "longer install"?

「インストール時間が長くなる」のは解決できる？

MAKEFLAGS="-j$(nproc)"...?

パラレルビルドでがんばれる…？

How to resolve "system change"?
「システムが変更される」のは解決できる？

- ✓ Container per app?
  アプリごとにコンテナーを作ればいいのでは？
- ✓ Package manager per app?
  アプリごとにパッケージマネージャーを用意すればいいのでは？
  - ✓ e.g.: Conda, Conan, vcpkg, ...
  - ✓ Homebrew!? (It uses Ruby!)
    Homebrewとか！？Ruby使っているし！

Proposal
提案

- ✓ Reduce maintenance cost for sustainable Ruby ecosystem
  持続可能なRubyエコシステムのためにメンテコストを減らさない？
  - ✓ Well maintained fat gems are useful for users but they have high maintain cost...
    メンテされているfat gemが便利なのはわかるけど大変なんよ。。。
  - ✓ Maintainers can use their time for others
    メンテナーは他のことに時間を使えるようになるよ
- ✓ Goodbye fat gem with auto preparation mechanism
  自動で準備してくれる仕組みでfat gemにさよならできない？

Without fat gem

fat gemをやめると

Happy!

...Really?

…ほんとに？

What do the RubyGems devs think?

RubyGems開発者はどう考えているの？

"Developing wheels for RubyGems"

RubyGems用のwheelsを開発しているよ

https://blog.rubygems.org/2025/03/19/february-rubygems-updates.html

...Really?

…マジ？

私はPyArrowのwheelをちょっとメンテしているけどマジ大変よ。。。

Wrap up
まとめ

Thanks fat gem!
Goodbye fat gem...?

ありがとうfat gem！
さよならfat gem…？

Let's discuss this!
相談しようぜ！

I'll be the followings:

- ✓ Around the venue（そこらへんをぶらぶら）
- ✓ @.bookstore in long breaks
  長い休憩のときはささださんの本屋さんで店員のお手伝いしているかも
  リーダブルコードのサイン会にもいるよ
- ✓ Day 1: Official party
- ✓ Day 2: Ruby Development Inc. Drinkup
- ✓ Day 3: Code Party（コード懇親会）